

KNN and AHC Algorithm for Text Summarization based on Text Clustering considering Feature Similarities

Taeho Jo
President
Alpha AI Research
Cheongju, South Korea
tjo018@naver.com

Abstract—In this research, we propose the KNN algorithm and the AHC algorithm which consider the feature similarities for automating the text summarization, using the text clustering. In the previous work, even if the KNN version was proposed as an approach to the text summarization, a domain should be presented as an input text tag, manually. In this research, text clusters are constructed based on their contents by the text clustering, and from a novice text, its summary is extracted by selecting a classifier which corresponds to its most similar cluster. The AHC algorithm and the KNN algorithm which consider the feature similarities are adopted respectively as the approaches to the text clustering and the text summarization. The goals of this research are to automate the text summarization and to improve the performance of both tasks, using the two algorithms which consider the feature similarities.

I. INTRODUCTION

The text summarization is defined as the process of extracting the essential part from a text as its summary. It is interpreted into a binary classification where each paragraph is classified into summary or non-summary. The process of text summarization is to partition an input text into paragraphs, classify each paragraph into one of the two categories, and extract ones which are classified into summary. The classification which is mapped from the text summarization is an instance of the domain dependent classification where a same entity may be classified differently depending on the domain. It is necessary to present the domain as well as the input text for summarizing it automatically.

This research is motivated by the necessity of defining domains by prior knowledge for designing the text summarization system. The text summarization is interpreted into a binary classification as mentioned above; the text summarization system is composed with independent binary classifiers as many as domains. The process of designing the text summarization system is to predefine domains by prior knowledge or subjectively and collect sample paragraphs domain by domain. One more motivation of this research is the empirical validations of the KNN algorithm and the AHC algorithm which considers the feature similarities as their better performances in the text summarization and the text clustering. This research is intended to automate the

domain definition by combining the text summarization with the text clustering.

The idea of this research is to apply the KNN algorithm and the AHC algorithm which consider the feature similarities to the text summarization and the text clustering which related to each other. In this research, we prepare the text collection where each text is associated with its own summary. The similarity metric between two numerical vectors which considers the feature similarities, and the KNN algorithm and the AHC algorithm are modified based on the similarity metric as the approaches to both tasks. The collection is segmented into clusters by the text clustering, and the text summarization system is implemented for each cluster independently of other clusters. A domain is automatically decided to an input text by its similarities with the cluster prototypes for selecting a corresponding classifier.

Let us mention some benefits from this research. The poor discrimination among sparse vectors is overcome by considering the feature similarities in computing the similarity between two numerical vectors. The performances in the text summarization and the text clustering are improved by adopting the modified versions of the KNN algorithm and the AHC algorithm. It does not require to define domains depending on prior knowledge or subjective views by automating it through the text clustering which relates to the text summarization. We do not need to present a domain by deciding automatically it based on the similarities of an input text with the cluster prototypes.

Let us mention some remaining tasks as the further discussions on this research. Only some features which are selected as the essential ones should be involved in computing the feature similarities for reducing the computation cost. We modify more advanced machine learning algorithms and deep learning algorithms into the versions which consider the feature similarities. We should implement the text summarization system which relates to the text clustering as a real program or a library. The text classification or the text clustering relates to with the text summarization for improving its performance.

II. PROPOSED WORKS

A. Encoding Process

This section is concerned with the process of encoding a text into a numerical vector. Words are used as features for encoding a text so, and some are selected from words in a corpus. The similarity between words is computed based on texts with their collocations and is used as the similarity between features. The similarities among features are considered for computing the semantic similarity between texts. In this section, we describe the process of encoding a text into a numerical vector and the process of computing the similarity between texts.

The process of encoding texts into numerical vectors is illustrated in Figure 1. The entire text collection is indexed into a list of words which are feature candidates. Some words are selected as the features by criteria among the feature candidates in the second step. The weights of the selected features in the text are computed as their relationships with the text, as elements of the numerical vector. Refer to [2] for studying the process and the selection criteria in detail.

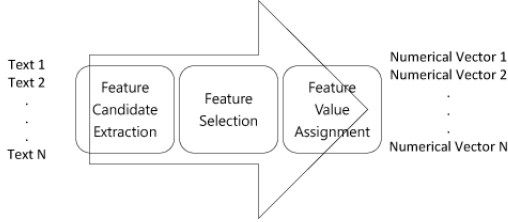


Figure 1. Process of Encoding Words into Numerical Vectors

The frame of computing the similarity between two numerical vectors is illustrated in Figure 2. The two d dimensional vectors and the d features are respectively notated respectively by $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_d]$, $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_d]$, and $f_1, f_2, \dots, f_d$. The feature similarity refers to the similarity between two features, which is notated by $\text{sim}(f_i, f_j)$. The feature value similarity is the similarity between two vectors, $\mathbf{x}$ and $\mathbf{y}$, such as cosine similarity. The similarity between words is computed by considering the two kinds of similarities.

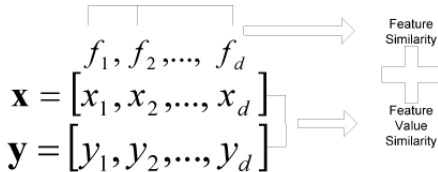


Figure 2. Frame of Computing Similarity between Numerical Vectors

Let us mention the process of computing the similarity between numerical vectors as the similarity between texts. The similarity between the features, f_i and f_j , is notated

by $\text{sim}(f_i, f_j) = s_{ij}$. The similarity between the features is computed by equation (1),

$$s_{ij} = \text{sim}(t_i, t_j) = \frac{2 \times \#(t_i \wedge t_j)}{\#(t_i) + \#(t_j)}, \quad (1)$$

where $\#(t_i \wedge t_j)$, is the number of texts which include both words, t_i and t_j , $\#(t_i)$ is the number of texts which include the word, t_i , and $\#(t_j)$ is the number of texts which include the word, t_j , and the similarity matrix to the d features is constructed as a $d \times d$ square matrix as follows:

$$S = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1d} \\ s_{21} & s_{22} & \dots & s_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ s_{d1} & s_{d2} & \dots & s_{dd} \end{pmatrix}.$$

The similarity between the two numerical vectors, $\mathbf{x}$ and $\mathbf{y}$, is computed by equation (2),

$$\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^d \sum_{j=1}^d s_{ij} x_i y_j}{d \|\mathbf{x}\| \|\mathbf{y}\|} \quad (2)$$

Equation (2) is used for computing the similarity between a novice text and a sample text in the KNN algorithm.

Let us make some remarks on the encoding process and the computation of the similarity between two texts. A text is encoded into a numerical vector by the feature candidate extraction, the feature extraction, and the feature value assignment. The similarity between features which are given as words is computed by equation (1). The similarity between numerical vectors which represent texts is computed by equation (2). In future, we consider computing the similarities among some features rather than all features for improving the computation speed.

B. Feature Similarity based KNN Algorithm

This section is concerned with the KNN algorithm which considers the feature similarities. The version was initially proposed as the approach to the text summarization by Jo in 2019 [3]. The similarity metric between numerical vectors which was described in the previous section is used for computing the similarity between a novice vector and a training example. The labels of its nearest neighbors are voted with their identical weights for decoding the label of a novice vector. This section is intended to describe the initial version of the KNN algorithm from which its variants are derived.

The process of computing the similarity between a novice item and a training example is illustrated in Figure 3. The labeled words which are gathered as samples are encoded into numerical vectors, and they are notated by a set $Tr = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$. A novice word is encoded into a numerical vector notated by $\mathbf{x}$. Its similarities with the numerical vectors which represent the sample words are computed by

equation (2), as $sim(\mathbf{x}, \mathbf{x}_1), sim(\mathbf{x}, \mathbf{x}_2), \dots, sim(\mathbf{x}, \mathbf{x}_N)$. Some training examples which are most similar as the novice input are selected as its nearest neighbors.

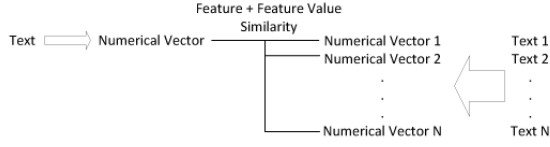


Figure 3. Similarities of Novice Input with Training Examples

In Figure 4, the process of selecting nearest neighbors by the ranking selection is illustrated. The training examples are sorted by the descending order of their similarities, after computing their similarities with a novice example. The training examples with their higher similarities with a novice example are selected as its nearest neighbors, as expressed in equation (3),

$$Nr = Select_k(Tr, \mathbf{x}), Nr \subseteq Tr \quad (3)$$

The set of nearest neighbors, Nr , is composed with elements as expressed in equation (4),

$$Nr = \{\mathbf{x}_1^{near}, \mathbf{x}_2^{near}, \dots, \mathbf{x}_k^{near}\} \quad (4)$$

The elements in the set, Nr , are used for voting their labels in the next step.

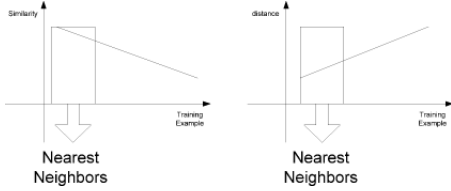


Figure 4. Selection of Most Similar or Least Distant Training Examples as Nearest Neighbors

The process of voting the labels of nearest neighbors for decoding the label of a novice word is illustrated in Figure 5. The predefined categories are notated by $c_1, c_2, \dots, c_m$, and the label of a nearest neighbor is notated by $c_j = label(\mathbf{x}_i^{near})$. The number of nearest neighbors which belong to the category, c_j , is notated by $Count(Nr, c_j)$. The label of the novice item, $\mathbf{x}$, is decided by equation (5),

$$label(\mathbf{x}) = \arg\max_{j=1}^m Count(Nr, c_j) \quad (5)$$

The process of deciding the label of a novice item by equation (5), is called voting.

Let us make some remarks on the KNN algorithm which considers the feature similarities. It computes the similarities of a novice item with the training examples. The training examples with their highest similarities with the novice item are selected as its nearest neighbors. The label of the novice item is decided by voting the labels of its nearest neighbors.

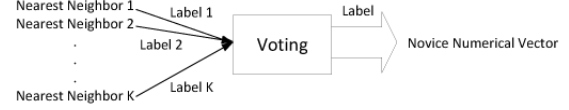


Figure 5. Voting of Labels of Nearest Neighbors for Deciding Label of Novice Input

The ranking selection is adopted for selecting the nearest neighbors from training examples, in this version.

The process of sampling paragraphs domain by domain is illustrated in Figure 6. Even if it is possible map the text summarization into an instance of text classification, a paragraph may be classified differently depending on the domain. Sample paragraphs which are labeled with summary or non-summary are gathered domain by domain. The text which is given as the input is tagged with a domain, and the classifier which corresponds to the domain is appointed. The sample paragraphs are encoded into numerical vectors in each domain.

C. System Architecture

This section is concerned with the text summarization system which adopts the KNN algorithm which considers the feature similarities. In Section II-B, we described the proposed version of KNN algorithm as the approach to the text summarization. It is viewed as a binary classification which classifies a paragraph into summary or non-summary. This section is intended to describe the text summarization system with respect to its function and its architecture.

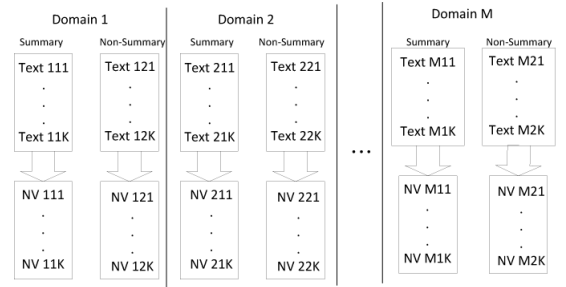


Figure 6. Preparation of Training Examples

The system architecture of the text summarization system is illustrated in Figure 7. The text partition module partitions the text which is given as the input into paragraphs. The encoder module encodes the paragraphs into numerical vectors by the process which is described in Section II-A. The similarity computation module computes the similarities of each paragraph with the sample paragraphs by the process which is described in Section II-A and selects ones with highest similarities as the nearest neighbors. The voting module votes the labels of nearest neighbors for decoding the label of the novice item.

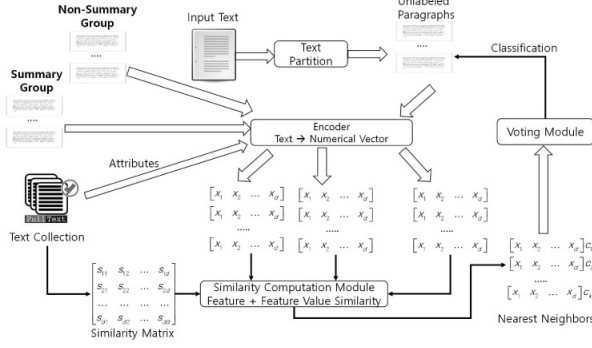


Figure 7. System Architecture

The execution flow of the text summarization system is illustrated in Figure 8. In each domain, sample paragraphs which are labeled with summary or non-summary are gathered. The input text is partitioned into novice paragraphs, and both sample paragraphs and novice paragraphs are encoded into numerical vectors. Each paragraph is classified into summary or non-summary, and there are two groups of paragraphs in the text: summary group and non-summary group. The paragraph which are classified into summary are extracted as the summary of the input text in this system.

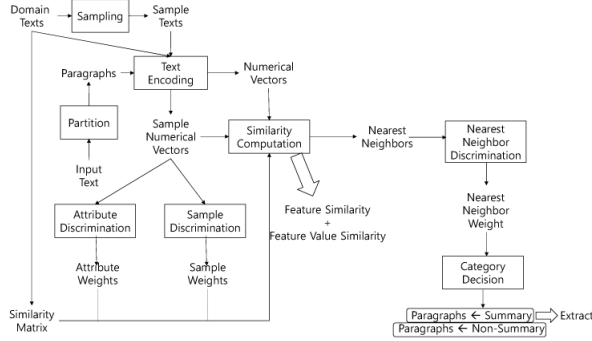


Figure 8. System Flow

Let us make some remarks on the system architecture and the execution process of the text summarization system which are presented in Figure 7 and 8. The text summarization is viewed into a binary classification of paragraphs, and we propose the similarity metric between two numerical vectors which is tolerant to the sparse distribution over them. The KNN algorithm is modified by defining the similarity metric between a novice paragraph and a sample paragraph, and the proposed version is applied to the text summarization. The system architecture and the execution flow which are provided by this research are needed for making the general design of the system. In the next research, we consider the detail design and the implementation of the system.

D. Connection with Text Clustering

This section is concerned with the connection of the text summarization with the text clustering. In the previous section, we mentioned the text summarization as an instance of domain dependent classification. The domains are automatically constructed from the text collection by clustering texts. The AHC algorithm which is proposed in [1] is used for implementing the text clustering. This section is intended to describe the text summarization system which is connected with the text clustering for automating the construction of domains.

The architecture of the text clustering system which was proposed in [1] is illustrated in Figure 9. The texts in the group are encoded into numerical vectors, and the AHC algorithm which is proposed in [1] is adopted as the approach. It starts with singletons as many as items, computes the similarities of all possible cluster pairs, and merge the cluster pair with its highest similarity into one cluster. The two steps are iterated until the number of clusters reach the desired number. We may consider replacing the AHC algorithm by its variants for improving the speed and the performance.

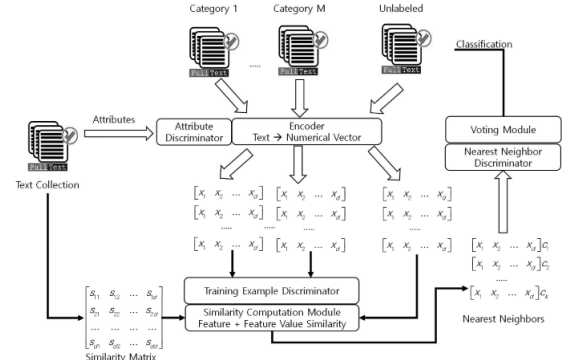


Figure 9. Text Clustering System

The connection of the text summarization with the text clustering is illustrated in Figure 10. The M domains are defined by clustering texts in a group, initially and automatically. A novice text is given as the input, and its domain, domain D , is decided by its similarities with domains. A classifier which corresponds to domain D is nominated and classify each paragraph in a novice text into summary or non-summary. The M text summarization systems are viewed as independent ones, each of which classifies each paragraph into one of the two categories.

The execution flow of text summarization which is connected with the text clustering is illustrated in Figure 11. Unlabeled texts are clustered into subgroups, each of which is a domain. Sample paragraphs which are labeled with summary or non-summary are generated from each domain. These sample paragraphs are provided for training the classifier in

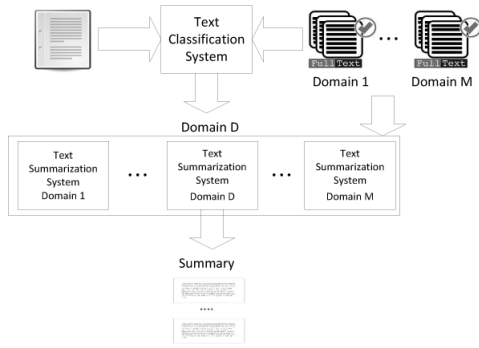


Figure 10. Text Summarization System connected with Text Clustering

each text summarization system. Each classifier is used for judging whether each paragraph in the input text is essential or not.

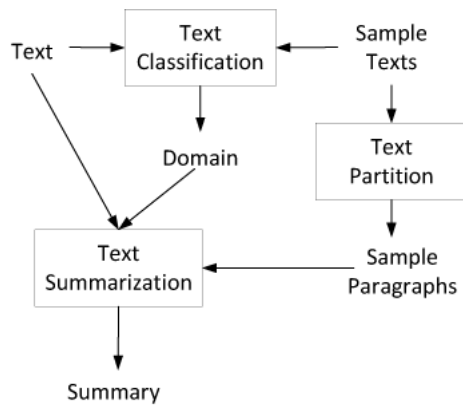


Figure 11. System Flow of Text Summarization connected with Text Clustering

Let us make some remarks on the connection of the text summarization with the text clustering. It is the process of segmenting a text group into subgroups based on their content-based similarities. The role of text clustering is to define the domains initially in the architecture of connecting the text summarization with the text clustering. In each domain, sample paragraphs are generated, and its corresponding classifier is trained with them. In the future research, we consider using the text summarization for clustering texts in the opposite direction.

REFERENCES

- [1] T. Jo, "Clustering Texts using Feature Similarity based AHC Algorithm", 5993-6003, Journal of Intelligent and Fuzzy Systems, 5, 2018.
- [2] T. Jo, "Text Mining: Concepts and Big Data Challenge", Springer, 2019.
- [3] T. Jo, "text summarization using Feature Similarity based K Nearest Neighbor", 13-21, AS Medical Science, 3, 2019.